

TSS 環境におけるオブジェクト指向型ユーザ・ インタフェース実現の試み†

末 永 正^{††} 景 川 耕 宇^{†††} 大 槻 説 乎^{††††}

最近、オブジェクト指向型システムが注目をあびている。しかし、これらのシステムは、ビットマップ・ディスプレイ等の特殊なハードウェア装置を前提にしており、種々雑多な端末が存在する汎用計算機の TSS 環境で実現した例は少ない。本論文では、FACOM OS IV/F4 MSP における TSS 環境において、既存のソフトウェアとの互換を保ちながら、ユーザ・インタフェースをオブジェクト指向に改めたシステムの実現方法について述べる。

1. ま え が き

我々が汎用計算機を使用するとき、TSS による会話処理を必要とすることが多い。計算機との会話は、端末機に対するコマンド入力とその応答という形で行われる。TSS の操作性は、メニュー形式による画面表示など、日々改善の途にあるが、今なお、そのコマンドの多くは、タイプライタ型端末を指向した形式になっている。この形式では、コマンド(指令)が、TSS における会話の第一声になっている。最近では、スクリーン型端末での画面(メニュー)操作による会話が可能になっているが、これさえも、何をするかという指令がメニューをたどる上での単位となっている。

これに対して、コマンドが取り扱うファイルなどの対象物(オブジェクト)を主体にして、計算機と会話する方式がある。この方式の一例は、画面エディタを用いたテキスト編集である。

ここでは、前者をコマンド指向型、後者をオブジェクト指向型と呼ぶことにする。行エディタと画面エディタの例からもわかるように、コマンド指向型よりもオブジェクト指向型の方が操作性に優れている場合が多い。コマンド指向型は、利用者にとって使いやすいくからではなく、システムにとって指令を処理しやすいこと、初心者向のガイド付の会話方式を実現しやすいことなどによって、ほとんどの TSS で採用されている。

最近では、いくつかのオブジェクト指向型のシステム利用環境が実現されている^{1),2)}。しかしながら、これらは、ビットマップ・ディスプレイ、およびポインティング・デバイスなどのハードウェア環境を前提にしたワークステーション上のものが多く、種々雑多な端末が存在する一般の TSS 下での実現例は少ない。一部に、内部制御をオブジェクトに基づいた方式に改めた例³⁾が見られるが、このユーザ・インタフェースは従来のコマンド指向型である。

本論文では、FACOM OSIV/F4 MSP における TSS 環境⁴⁾のもとで、オブジェクト指向型のユーザ・インタフェース実現を試みた会話型システム(TSS/BTF: Brief Touch Facility)、およびその実現上の問題等について述べる。ここでは、コマンド処理に際して、smalltalk¹⁾と同様に、オブジェクトにメッセージを送るという方式によって実現している。

2. システム環境

2.1 TSS の制御概要

TSS では、端末、およびシステム内部の動作状態を監視する会話型モニタ(以後、TMP: Terminal Monitor Program と呼ぶ)のもとで、ユーザとの会話が行われる。通常、これらの会話は、ユーザからのコマンド投入、およびそれに対するシステムからの応答という形で進められる。図1に、TMP による会話処理の概略を示す。

TSS 端末から入力されたコマンド(指令)は、通信プログラムを通して、TMP に渡る。OSIV/F4 MSP での TSS のコマンド(以後、TSS コマンドと書く)の入力(コマンド文)は、一般に、次の形式をしている。

コマンド名 対象物名 パラメータ (1)

† Experience in Implementing Object-Oriented User-Interface under TSS by TADASHI SUENAGA (Educational Center for Information Processing, Kyushu University), KOU KAGEKAWA (Computer Center, Kyushu University) and SETSUKO OTSUKI (Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology).

†† 九州大学情報処理教育センター

††† 九州大学大型計算機センター

†††† 九州工業大学情報工学部知能情報工学科

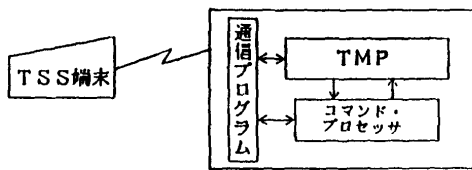


図1 TMPによる会話処理の概略

Fig. 1 TSS conversation controlled by TMP.

TMPでは、コマンドに対応するプログラム(コマンド・プロセッサ)を呼び出す。コマンド・プロセッサは、パラメータの指示に従って対象物进行操作し、端末に結果を表示する。TMPは、コマンド・プロセッサの終了を待って、次のコマンド入力をユーザ(端末)に要求する。

ところで、会話手段としての端末には、様々な種類がある。例えば、

- ①画面単位での入出力が可能なもの(フルスクリーン型)
- ②ライン表示形式のディスプレイ(CRT型)
- ③印字形式のもの(タイプライタ型)

などがある。①は、画面単位での入出力であり、メニュー方式で会話を進めるツールが用意されている⁵⁾。これに対して、②と③は、従来から用いられている1行単位の入出力である。しかしながら、(1)のコマンド形式は、端末種類にかかわらず、すべての端末の基本入力形式である。

2.2 TSSにおけるオブジェクト

TSSにおける処理は、コマンド名に対応するコマンド・プロセッサによって、処理される。各コマンド・プロセッサでは、処理すべき対象物の種類、および指定方法(コマンド構文)があらかじめ決まっている。このような環境では、コマンド・プロセッサが扱う対象物をコマンドのオブジェクトとし、オブジェクト指向型の制御方式に変更することも可能である。オブジェクトとしては、単一の対象物だけでなく、その集合もオブジェクトになりうる。

OSIV/F4 MSPでのTSSのオブジェクトには、主として、次のようなものが考えられる。

i) ファイル

1つのファイルの中がさらに別の単位(メンバ)に分割される(分割型ファイル)場合、各メンバもそれぞれ1つのオブジェクトである。

ii) ジョブ

ユーザに帰属するジョブであり、TSSのジョブ(TSSセッション)とバッチジョブがある。

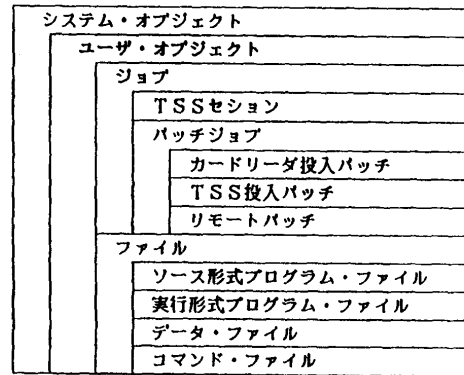


図2 オブジェクトのクラス構成の例

Fig. 2 An example of class organization.

2.3 オブジェクト指向概念とTSS環境の整合性

オブジェクト指向の最も重要な概念として、①オブジェクト、②メッセージ、および③クラスがある。これらをTSS環境において適用した場合、①と②に対しては次のような対応が考えられる。

- オブジェクト => コマンドの対象物
- メッセージ => コマンド名 パラメータ

ただし、コマンド自体も別のコマンドによって作成されるオブジェクト(ファイル)であるため、ファイルの変化に伴って、メッセージも変化する可能性がある。

また、クラスは、

- オブジェクトの性質の抽象化であること。
- 階層構造をなすこと。
- 下位クラスは上位クラスの性質を継承すること。

の特徴をもつことから、図2のようなクラス構成が考えられる。

ここでのクラスの性質は、オブジェクトが受け取ることのできるメッセージの種類、およびメッセージの取扱い方法(メソッド)によって決定される。しかしながら、既存のOSのもとでは、メッセージと密接な関係にあるTSSコマンドも種々の独立したソフトウェアによって動的に変更される。したがって、存在するコマンド・ファイルの全TSSコマンドに対して、そのコマンド文、つまりメッセージとオブジェクトの関係を解釈しなければならない。このような解釈は、事実上、不可能に近いので、図2のようなクラスを一元的に管理するには無理がある。

このような状況では、オブジェクトからクラスの構成要素(インスタンス)を簡単に定義できるようなクラス構成が必要である。

2.4 TSS 下でのシステム設計の条件

TSS では、多数のユーザがシステムを共有して使用するため、会話型モニタは、以下の条件を満足している必要がある。

- i) ユーザ間での処理の独立性、およびデータの保全性が保障されると同時に、他ユーザのファイルも利用できること。
- ii) 会話型モニタ自身によって使用されるシステム資源 (CPU, メモリなど) ができるかぎり少ないこと。

また、会話処理としての性格から、

- iii) ユーザ固有のコマンド作成が可能なこと。
- iv) 入力コマンドの解釈、呼出、および後処理が高速であること。

の条件もある。さらに、ソフトウェア互換性の立場から、

- v) 作成されたユーザ・プログラムが他の会話型モニタ環境下でも動作できること。

が保障されなければならない。その他に、初心者から熟練者までの幅広い利用者層からなる計算センタのような場合、

- vi) 入力方法が分かりやすく、しかも、双方にとって、会話が円滑であること。

も重要である。

3. TSS/BTF の特徴

TSS/BTF (以下、単に BTF と書く) は、TSS 環境のもとで、プログラム実行に関する制御方式を変更することなく、ユーザ・インタフェースだけをオブジェクト指向に改変するシステムである。その実現方法は、TMP のもとで動作する新たな会話型モニタを作成し、そのなかで従来のコマンド制御とのインタフェースをとる方式である。したがって、既存のソフトウェアをそのまま利用でき、しかも BTF 環境下で作成したソフトウェアも従来のシステム下に移行できる。このことは、実現の容易性、ソフトウェアの双方向互換の点で極めて重要である。

ところで、BTF において、オブジェクト指向概念は次のように実現している。

i) オブジェクト

オブジェクトは、TSS コマンドが操作対象にしている計算機資源の中で、ユーザがアクセスできるものであり、ここでは、ジョブ、ファイル、およびこれらの集合である。

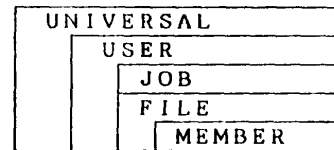


図 3 BTF におけるクラス構成
Fig. 3 Class organization in BTF.

ii) メッセージ

メッセージには、BTF がオブジェクトを管理するための内部メッセージ (BTF メッセージ) と、TSS コマンドを呼び出すためのメッセージ (TSS コマンド・メッセージ) がある。オブジェクトに送られた TSS コマンド・メッセージは、あらかじめ定義された変換メソッドに従って、コマンド文になる。

iii) クラス

2.3 節のクラス構成 (図 2) において用いたクラス性質、①オブジェクトの受け取れるメッセージ、②メッセージの取扱い方法 (メソッド) のうち、①は、メッセージによって呼び出される TSS コマンド自身にオブジェクトの正当性の判断機能が備わっているため、クラス構成の性質から除外しても事実上問題はない。

したがって、ここでは、オブジェクトの種類、および性質②から分類した単純なクラスを構成する。図 3 に示すように、BTF は、5つのクラスを設け、MEMBER を最下位のクラスにする。MEMBER のインスタンスは、分割型ファイルのメンバである。FILE は、MEMBER のスーパークラスであり、全ファイルがそのインスタンスである。ただし、ユーザには、MEMBER と FILE をあたかも同レベルのクラスのように見せ、操作時の単純化を図っている。USER は、JOB および FILE のスーパークラスであり、そこでは、ユーザ固有のメッセージ変換方式 (メソッド) を定義できる。UNIVERSAL は、BTF のルートクラスであり、全ユーザのデフォルト・メソッドを定義している。ユーザ・インタフェースを簡単にするために、UNIVERSAL や USER 以外のクラス、および各オブジェクトでは、メソッド定義機能をもたない。なお、クラスもオブジェクトの一種であり、メッセージを受け取ることができる。

BTF は、オブジェクト指向によるユーザ・インタフェースの改善を主目的にしている。したがって、ク

ラス構成は必要最小限にとどめ、ユーザがクラスの階層構造をあまり意識せずすむようにしている。また、オブジェクトからクラスのインスタンスを再定義する面からも、単純なクラス構造の方が望ましい。

一方、実際の会話に関するユーザ・インタフェースでは、オブジェクトをメニュー形式で表示するとともに、メッセージ入力方式を改善する。以下に、おもなユーザ・インタフェースを記述する。

3.1 オブジェクト表示

BTF は、3つのコマンド・プロセッサからなり、いずれか1つのコマンド・プロセッサによって起動される。これらのコマンドは、表示するオブジェクトのクラス(JOB, FILE, MEMBER)に対応しており、同一クラスに属するオブジェクトを選択的に切り出して、サブクラスを生成することができる。

各クラスの起動コマンドは、次の形式で入力する。

i) FLIST パラメータ

クラス FILE のサブクラスを生成し、そのインスタンスを表示する (図 4)。

ii) MLIST 分割型ファイル名 パラメータ

クラス MEMBER のサブクラスを生成し、そのインスタンスを表示する (図 5)。

iii) JLIST パラメータ

クラス JOB のサブクラスを生成し、そのインスタンスを表示する (図 6)。

BTF では、オブジェクトを番号によって管理しており、その識別番号をオブジェクトの先頭に表示する。ただし、フルスクリーン型端末の場合には、オブジェクトとの対応づけが画面上的入力位置によって可能なことから、図7のように、オブジェクト番号を省略した形式で表示する。

ところで、オブジェクトには、それぞれに固有の性質(プロパティ)があり、次のような属性や状態をも

クラス名	システム・レベル	ファイル名の第1修飾子	オブジェクト識別子	オブジェクト数
FILE	V20/L01	PREFIX(A70090A)	86/12/01	ID(F1)
1	EXAMPAAA	CNTL	PS	8
2	EXAMPBBB	DATA	PS	9
3	EXAMPCCC	CNTL	PS	10
4	EXAMPDDD	DATA	DA	11
5	EXAMPEEE	PORT	PS	12
6	EXAMPFFF	CNTL	PS	13
7	EXAMPFFF	PORT	PS	14
				15
				16
				17
				18
				19
				20
				CONTINUE

図 4 クラス FILE のサブクラス表示例
Fig. 4 An example of FILE subclass.

分割型ファイル名	ユーザ名 (*印が付いたものは、あるユーザの別名(alias)であることを示す)
MEMBER - V20/L01	A70090A.CMDPROC.CLIST
1	ALLOC 5 ATNAPLA 9 ATNSPL 13 DSCAN 17 DSTSTATUS
2	*ALLOCATE 6 ATNEASP 10 AXEL 14 *DSL 18 EASYSPPSS
3	APL 7 ATNFCMP 11 DEF 15 DSLIST 19 ENDQ
4	APLA 8 ATNSGV 12 *DSC 16 *DST 20 FCMP

図 5 クラス MEMBER のサブクラス表示例
Fig. 5 An example of MEMBER subclass.

ジョブ名	ジョブ番号	システム内の状態	ジョブ依頼者
JOB	V20/L01	USER (A70090A)	86/12/01
1	A70090A#(J0052)	E	8 A70090A5(J1102) WE
2	A70090A2(J0073)	P	9 A70090A8(J1103) WE
3	A70090A3(J0121)	VP	10 A70090A7(J1104) WE
4	A70090A1(J0122)	VP	11 A70090A5(J1125) WE
5	A70090A2(J0153)	E	12 A70090A8(J1341) WE
6	A70090A3(J0166)	E	13 A70090AX(J1447) WE
7	A70090A4(J1010)	E	END

図 6 クラス JOB のサブクラス表示例
Fig. 6 An example of JOB subclass.

つ。

i) ファイル

属性: ファイルの所有者 (ファイル名の第1修飾子)

ファイルの特性 (レコード長, レコード形式など)

状態: アクセス日 (作成/参照/更新)

容量 (割当量, 使用量など)

存在場所 (ディスクのボリューム名)

ii) メンバ

属性: レコード数 (ソース形式の場合), 他メンバとの関係

状態: 更新日時 (ソース形式の場合)

iii) ジョブ

属性: ジョブ依頼者

状態: 実行状況 (実行待ち, 出力待ち, CPU 時

```

***>
----- FILE ----- V20/L01 -- PREFIX(F0090 ) - 85/01/09 - ID(F1) -- 1/ 18
NAME TYPE ORG INPUT FIELD
EXAMPAAA CNTL PS
EXAMPBBB DATA PS
EXAMPCCC CNTL PS
EXAMPDDD DATA PS
EXAMPEEE FORT PS
EXAMPFFF CNTL PS
EXAMPFFF FORT PS
EXAMPHHH PLI PS
EXAMPIII PLI PS
EXAMPJJJ LOAD PO
EXAMPKKK PLI PS
EXAMPLLL FORT PS
EXAMPHHH DATA PS
EXAMPNNN BASIC PS
EXAMPOOO CNTL PS
EXAMPOOO FORT PS
EXAMPOOO LOAD PO
EXAMPOOO PLI PS
** END **

DISP=>PF6,FRVD=>PF8,BACK=>PF7,REPF=>PF5,INCF=>PF9, SORT=>PF4, SVAP=>PF2, TERN=>PF3
    
```

図 7 フルスクリーン型端末でのオブジェクト表示例
Fig. 7 A display on a full-screen terminal.

間、経過時間など)
負荷環境 (システム内
ジョブ数、実行多重
度など)
BTF では、オブジェクト・プロパ
ティも管理しており、プロパティに
よる操作 (例えば、サブクラスの生
成、オブジェクトの検索、ソーティ
ングなど) を可能にしている。図 8
に、ファイルに関するオブジェ
クト・プロパティの表示例を示す。

3.2 メッセージ入力

オブジェクトに送られるメッセ
ージには、TSS コマンドに変換される
TSS コマンド・メッセージと、BTF
内部で、オブジェクトを管理するた
めに使用するメッセージ (BTF メ
ッセージと呼ぶ) がある。BTF メ
ッセージは、“.” (ピリオド)、また
は“/” (スラッシュ) で始まってお
り、TSS コマンド・メッセージと
は容易に区別できる。表 1 に、おも
な BTF メッセージを示す。

BTF では、通常、メッセージを、
オブジェクト番号 メッセージ
の形式で入力する。この場合、番号で指定したオブ
ジェクトに、メッセージを送ることを意味する。オブ
ジェクト番号が 0 であれば、表示しているサブクラスに
対するメッセージとなり、その中のすべてのオブジェ

NO	FILE NAME	TYPE	ORG	RF	LRECL	BLKSZ	EX	SPACE	%	CREATE	VOLUME
1	EXAMPAAA.S5912	CNTL	PS	F	80	3120	1	13	**	800408	PUB011
2	EXAMPBBB.A5911	DATA	PS	F	80	3120	4	78	**	800110	PUB015
3	EXAMPCCC	CNTL	PS	F	80	400	1	13	**	790327	PUB002
4	EXAMPDDD	DATA	PS	F	80	3120	1	39	**	800317	PUB007
5	EXAMPEEE	FORT	PS	F	80	400	7	91	86	800213	PUB014
6	EXAMPFFF.X5802	CNTL	PS	F	80	3120	1	13	**	800402	PUB005
7	EXAMPFFF	FORT	PS	V	255	3120	1	39	34	800401	PUB006
8	EXAMPHHH	PLI	PS	F	80	400	1	13	**	790323	PUB004
9	EXAMPIII	PLI	PS	F	80	400	1	13	**	790316	PUB008
10	EXAMPJJJ.W5710.D12	LOAD	PO	U	-	13030	1	117	**	790316	PUB007
11	EXAMPKKK	PLI	PS	F	80	400	1	689	84	790314	PUB008
12	EXAMPLLL	FORT	PS	F	80	400	1	13	**	791220	PUB013
13	EXAMPNNN	DATA	PO	F	80	3120	1	117	**	781227	PUB004
14	EXAMPNNN	BASIC	PS	V	137	3120	1	13	**	800502	PUB007
15	EXAMPOOO	CNTL	PS	F	80	3120	1	13	**	800825	PUB005
16	EXAMPOOO	FORT	PS	F	80	400	1	26	**	800822	PUB008
17	EXAMPOOO	LOAD	PO	U	-	13030	1	689	21	800825	PUB017
18	EXAMPOOO	PLI	PS	F	80	400	1	52	75	800823	PUB001
19	EXAMPXXX	FORT	PS	F	80	400	1	47	**	840120	PUB111
20	EXAMPXXX	LOAD	PO	U	-	6144	1	94	**	840120	PUB112

2 行目の記号の意味は以下のとおり

- NO : オブジェクト番号
- FILE NAME: ファイル名から第 1 修飾子と最終修飾子を除いたもの
- TYPE : ファイル名の最終修飾子
- ORG : ファイル編成 (PS: 順編成, PO: 分割型編成など)
- RF : レコード形式 (F: 固定長, V: 可変長, U: 不定長)
- LRECL : レコード長 (バイト単位)
- BLKSZ : ブロック長 (バイト単位)
- EX : ファイルが占めるディスク上の連続領域の数 (最大 16)
- SPACE : ファイルが占める物理スペース量 (キロバイト単位)
- % : 物理スペース量の有効使用比率 (百分率, ** は 100%)
- CREATE : ファイルの作成日 (年月日)
- VOLUME : 存在するディスクのボリューム名

図 8 ファイルに関するプロパティの表示例
Fig. 8 Properties of file objects.

クトが同じメッセージを受け取るのと同じ効果があ
る。また、オブジェクト番号を省略すると、クラス
USER に対するメッセージとなる。この時のメッセ
ージが TSS コマンド・メッセージの場合、メッセ
ージがそのまま TSS コマンド文として処理される。

なお、フルスクリーン型端末の場合には、オブジェ

クト表示の右 (図7において, 4行目以降の右側空白部分) にメッセージを入力する. 表示されているオブジェクト全部 (サブクラス) を対象にするときには, 画面最上段にメッセージを入力する.

3.3 メソッド定義

ユーザは, メッセージから TSS コマンドに変換する方法を全オブジェクト共通のメソッドとして, 定義できる. その定義は,

メッセージ・キー ==> コマンド名 仮パラメータ (2)

のように指定する. ここで, メッセージ・キーは, 変換を行う際のキーとなる変数名であり, ユーザの入力メッセージを空白で区切って,

メッセージ・キー パラメータ1 パラメータ2... (3)

のように分解したときの先頭の文字列と対応する.

(2)の仮パラメータには, メッセージを受け取ったオブジェクトの名前を挿入する位置 (&0) のほか, 実パラメータ ((3)のパラメータ1, パラメータ2, ...) を &1, &2, ... 等として, 次の方法でその挿入位置を指定できる.

&1, &2, ...: 実パラメータの位置を示す.

&restPARAM: &1, &2, ... で指定した実パラメータ以外

例えば, テキスト編集用 TSS コマンド (EDIT) に対するクラス UNIVERSAL のメソッド定義は次のようになっている.

EDIT ==> EDIT &0 &restPARAM

この場合, メッセージを受けたオブジェクトの名前がコマンド EDIT の第1パラメータに指定されたことになる.

4. TSS/BTF の制御方式

BTF は, 既存ソフトウェアの変更を伴わないように, TMP のもとで, 通常のコマンド・プロセッサとして動作する方式を採る (図9).

BTF の起動には, 表示するオブジェクトの種類に応じたコマンド (FLIST, MLIST, JLIST) を用いる. BTF 下では, これらの起動コマンドも他の TSS コマンドと同様に扱われる. しかし, 使用する記憶領域の節約などから, BTF が階層構造をなすことはなく, 再帰的に起動されるコマンドは, 生成したサブクラスの管理を既に存在する BTF に依頼して終了する. この場合, BTF は, 複数のサブクラスを管理すること

表1 おもな BTF メッセージ
Table 1 Typical BTF messages.

分類	メッセージ	機能
起動	FLIST MLIST JLIST	ファイルのサブクラス生成 メンバのサブクラス生成 ジョブのサブクラス生成
終了	.END	サブクラス取消
スクロール	.TOP .BOTTOM .UP .DOWN	先頭からのオブジェクト表示 最後のオブジェクト表示 上方向のスクロール 下方向のスクロール
検索	/文字列 .FIND	オブジェクトの名前による検索 プロパティによるオブジェクト検索
表示	PROPERTY	プロパティの表示モード制御
ソート	.SORT	オブジェクトの並べ替え表示
選択	.SELECT .EXCLUDE	インスタンスの抽出 インスタンスの除外
再設定	.RENEW	オブジェクト情報の再設定
切替	.SWAP	アクティブ・サブクラスの切替
定義	.DEFINE	変換メソッドの定義

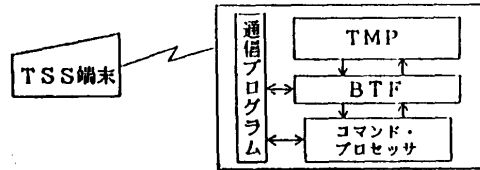


図9 BTF の動作環境

Fig. 9 Operation environment of BTF.

になる. ただし, 端末には, 同時には1つだけのサブクラスを表示し, それに対するメッセージ管理を行う (端末に表示しているサブクラスをアクティブ・サブクラスと呼ぶ). 表示外のサブクラスは, 非アクティブ・サブクラスとして, メッセージ送付の対象にはならないが, BTF メッセージによる切替操作でいつでもアクティブ・サブクラスに変更することができる.

BTF の制御モジュールは, 機能別に分かれており, 図10に示す6つの構成要素からなる.

①から⑥までの各構成要素は, 互いに処理が独立しており, この間の制御もすべて BTF マネージャを介したメッセージによって行われる. 以下に, 各構成要素についての概略を述べる.

4.1 端末入出力部

端末入出力部は, 端末との入出力を受け持ち, 端末特性を利用した表示方法を用いる. ここでは, フルス

クリーン型とタイプライタ型の2種類を用意している。

フルスクリーン型では、画面単位の入出力が可能であるため、画面上の入力位置によって、オブジェクトとの対応づけを行う。また、一度に複数のメッセージ入力も可能である。

タイプライタ型では、オブジェクトとの対応づけのため、メッセージの直前に送付先のオブジェクト番号が指定されている必要がある。また、入力を1行単位にしているため、一度に複数のメッセージを受け付けることはできない。

なお、入力されたメッセージは、オブジェクト番号を伴ってメッセージ解釈部に渡る。また、この端末入出力部は、各構成要素間で行きかう伝達メッセージがなくなった時、BTF マネージャからの指示によって動作する。

4.2 メッセージ定義部

メッセージ定義部は、オブジェクト管理部から送られた“.DEFINE”メッセージによって動作し、そのパラメータで指定された定義用テキスト・ファイルを解釈する。記述に間違いがなければ、そのメッセージ変換方式をクラス USER のメソッドとして登録する。一度登録した変換メソッドは次の登録まで有効であり、TSS セッションを終了しても保存される。

4.3 メッセージ解釈部

メッセージ解釈部は、メッセージが BTF メッセージ(表1)の場合には、そのままオブジェクト管理部に送る。TSS コマンド・メッセージの場合には、クラス USER または UNIVERSAL の変換メソッドに基づいて、メッセージをコマンド文に変換する。例えば、

```
ALC => ALLOC DATASET(&0) FILE(&1)
      &restPARM
```

の定義があるとき、図4で示されるアクティブ・サブクラスに対して、

```
2 ALC FT01F001 SHR REUSE
```

のメッセージを受け取ると

```
&0=EXAMPBBB.DATA
```

```
&1=FT01F001
```

```
&restPARM=SHR REUSE
```

として

```
ALLOC DATASET(EXAMPBBB.DATA)
FILE(FT01F001) SHR REUSE
```

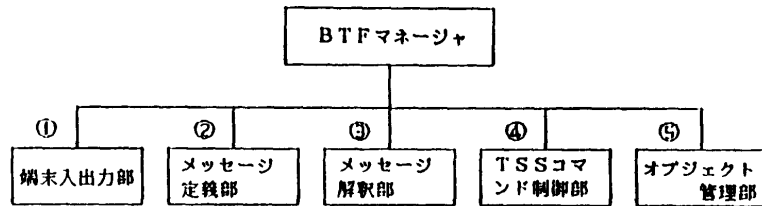


図10 BTFの構成図

Fig. 10 Software configuration of BTF.

のコマンド文に変換する。

指定したオブジェクトがサブクラス(オブジェクト番号0)の場合には、そのインスタンスを&0の位置に並べたコマンド文に変換する。

クラス UNIVERSAL における標準的な変換メソッドでは、メッセージ・キーをコマンド名に、パラメータをそのままコマンドの第2パラメータに変換する。このときの変換過程は次のようになる。

オブジェクト番号(仮に n とする) メッセージ

↓

オブジェクト番号 メッセージ・キー パラメータ

↓

オブジェクト番号 コマンド名 パラメータ

↓

コマンド名 番号 n のオブジェクト名 パラメータ
ただし、クラス USER に送られた(オブジェクト番号を指定しない)メッセージの場合には、このような変換メソッドはなく、そのままコマンド文になる。

なお、特殊なメッセージとして、“=”(イコール)がある。これは、複数のオブジェクトに対して、同じメッセージを繰り返して入力する場合に利用される。そのために、メッセージ解釈部は、送られたメッセージを次の解釈まで待避している。

最後に、変換によって生成したコマンド文を TSS コマンド制御部に送る。

4.4 TSS コマンド制御部

TSS コマンド制御部は、メッセージ解釈部が形式変換によって生成したコマンド文に対して、該当するコマンド・プロセッサを呼び出す。コマンド・プロセッサを起動した後は、単にその終了を待つ。

呼び出したコマンド・プロセッサがオブジェクトを変更するもの(DELETE, RENAME, および CANCEL)であれば、その実行結果に従って、メッセージを受け取ったオブジェクト、またはサブクラスの管理情報を変更する。これは、オブジェクト情報の再設定を指示するメッセージをオブジェクト管理部に送るこ

とによって行うが、アクセス日などのプロパティの変化までは反映されない。また、この変更は、アクティブ・サブクラスに対してのみ有効で、同一オブジェクトが非アクティブ・サブクラスのインスタンスであっても、その制御は及ばない。

なお、実行する TSS コマンドがオブジェクトを消去するもの (DELETE コマンド (ファイル) または CANCEL コマンド (ジョブ)) の場合には、オブジェクトの指定ミスを防止する意味で、再確認の入力をユーザに要求する。

4.5 オブジェクト管理部

オブジェクト管理部は、メッセージが “.DEFINE” であれば、メッセージ定義部へ送り、それ以外であれば、メッセージに従ってオブジェクトを操作する。オブジェクト操作には、大別して、以下のものがある。

i) サブクラスの生成

起動コマンド (FLIST, MLIST または JLIST) のパラメータに従って、該当するオブジェクトを抽出し、サブクラスを作成する。パラメータには、以下の選択条件を指定できる。

(a) FLIST

- ファイルの所有者 (ファイル名の第1修飾子)
- ファイル名の部分文字列
- 存在場所 (ボリューム名の部分文字列)
- ファイル編成 (分割型編成、順編成など)
- ファイル属性 (レコード形式など)
- アクセス情報 (作成日の範囲など)

ファイル名やボリューム名の場合、任意の文字列を示すワイルド・カードを指定できる。

(b) MLIST

- メンバ名の部分文字列
- 他メンバとの関連
- アクセス情報 (作成日時範囲など)

メンバ名には、任意の文字列を示すワイルド・カードを指定できる。

(c) JLIST

- ジョブの依頼者
- 実行状態 (実行待ち、出力待ちなど)

ii) オブジェクト管理情報の更新

TSS コマンドの実行によって、オブジェクトのプロパティが変更された時、その管理情報を最新のものに更新する。

iii) サブクラスの取消

複数のサブクラスから指定したものを制御対象から外す。サブクラスが1つも存在しなくなる場合には、BTF 自身も終了する。

iv) アクティブ・サブクラスの切替

非アクティブ・サブクラスをアクティブ・サブクラスに切り替える。

v) サブクラスの変更

アクティブ・サブクラスに対して、そのインスタンスを変更する。その変更方法には、選択 (SELECT) と排除 (EXCLUDE) がある。

vi) オブジェクトの表示形式

プロパティ表示、スクロール、ソーティングなどがある。

5. 実験と考察

BTF によって、ユーザは、オブジェクト名をいちいち記憶する必要がなく、メニューで示されるオブジェクトに対して、コマンドに対応したメッセージを送付するだけでよい。このような簡便さから、BTF のプロトタイプをユーザに公開⁶⁾して以来、全ユーザの約7割が BTF のもとで会話をを行っている。

一般的に、コマンド指向型のメニューによる入力方式では、初心者向けには、不慣れたコマンドを意識しないですむ利点がある。反面、かなり慣れた利用者にとっては、メニューをたどるのがかえって煩わしいことがある。BTF では、オブジェクト指向型のメニュー指示であり、まわりくどい入力を要しないため、初心者から熟練者までの幅広い操作性向上を図ることができた。

BTF は、既存の TSS 制御方式をそのまま利用し、ユーザ・インタフェースの変更だけでシステムを構築しているため、コマンド・プロセッサによって変化するオブジェクトをすべて把握することができない。それを行うには、TSS コマンドが終了するたびに、管理しているサブクラスに含まれるべき全オブジェクトを再調査しなければならない。しかしながら、その調査コストを考えると、現実的ではない。このような表示上の矛盾に対して、ここでは、メッセージによってどのオブジェクトが影響されるかは、入力するユーザに分かっているものとして、ユーザにその調整を任せることとした。このことは、会話型モニタのオーバヘッドが効率上重要である TSS 環境での実現の限界であり、今後の課題でもある。

6. む す び

以上、オブジェクト指向のユーザ・インタフェースを通常の TSS 環境のもとで実現した会話型システム (TSS/BTF) について述べた。これは、コマンドの対象物 (オブジェクト) を端末に表示し、そのオブジェクトに対して、メッセージ (例えば、コマンド名とオプション) を指定するという方式である。当然のことながら、オブジェクト名の入力が簡素化されるため、従来のコマンド指向型に比べて、入力タッチ数が大幅に減少する。

一般に、ユーザが取り扱うオブジェクトは、計算機の利用とともに変化し、しかも、その数は増加しがちである。これに対して、使用するコマンドの方は、比較的限られており、初心者の方は少なく、慣れた利用者でもせいぜい 20 個くらいである。コマンドの利用統計⁷⁾をみると、上位 20 のコマンドで全体の約 80% を占める。これらのことから、TSS の会話環境においても、コマンド指向型の入力方式より、オブジェクト指向型の入力方式の方が使いやすいといえる。

BTF は、その考え方として、SMALLTALK⁸⁾ に影響を受けており、オブジェクト、メッセージ、クラスなどの基本概念を導入している。ただし、ここでは、TSS 環境下におけるいくつかの制約から、クラス構成を単純化する必要があった。

BTF は、PL/I に似たシステム記述言語で書かれており、約 2 万ステップのプログラムである。その実行に際して、常時必要な主記憶量は、BTF マネージャと各構成要素のメッセージ受信部分で占められるプログラム域 (23 キロバイト)、および生成するサブクラスのインスタンス数に比例する作業域 (1 インスタンスあたり約 100 バイト) の合計である。

BTF をユーザに公開して 1 年余りが経過するが、操作性、およびメッセージの応答性に関するユーザの不満はほとんどでない。

謝辞 日頃ご指導と有益なご助言をいただいている九州大学工学部牛島和夫教授、ならびに九州大学大型計算機センター相良節夫教授に感謝いたします。また、TSS/BTF システムの作成にあたって協力していただいた九州大学大型計算機センター業務掛平野広幸氏に謝意を表します。

参 考 文 献

- 1) Goldberg, A.: *Smalltalk-80—The Interactive*

Programming Environment, Addison-Wesley, Reading, MA (1983).

- 2) アップル(社)マニュアル: Macintosh Plus (1983).
- 3) Laff, M. R. and Hailpern, B.: SW2—An Object-based Programming Environment, *SIGPLAN Not.*, Vol. 20, No. 7, pp. 1-11 (1985).
- 4) 富士通(社)マニュアル: FACOM OSIV/F4 MSP TSS コマンド文法書 78SP-1341-1 (1983).
- 5) 富士通(社)マニュアル: FACOM OSIV/F4 MSP PFD 使用手引書 プログラム開発機能編, 78SP-3101 (1983).
- 6) 末永, 平野: TSS コマンド入力支援システム (TSS/BTF) 第 2 版について, 九州大学大型計算機センター広報, Vol. 19, No. 1, pp. 4-104 (1986).
- 7) 富山, 景川: 利用状況統計からの一考察, 全国共同利用大型計算機センター研究開発論文集, No. 7, pp. 122-125 (1985).
- 8) Goldberg, A. and Robson, D.: *Smalltalk-80—The Language and Its Implementation*, Addison-Wesley, Reading, MA (1983).

(昭和 62 年 1 月 8 日受付)

(昭和 62 年 7 月 9 日採録)

末永 正 (正会員)

昭和 24 年生。昭和 47 年九州大学工学部電子工学科卒業。九州大学大型計算機センターを経て、現在、九州大学情報処理教育センター助手。オペレーティング・システムの効率、およびマンマシン・インタフェースに関心がある。人工知能学会会員。



景川 耕宇 (正会員)

昭和 15 年生。昭和 37 年東京大学工学部応用物理学科 (数理工学専攻) 卒業。昭和 39 年同大学院修士課程修了。(株)三永通信、(株)新日本パイプを経て昭和 42 年より九州大学勤務。現在九州大学大型計算機センター講師。主な研究分野: 計算機システムの性能評価。



大槻 説乎 (正会員)

1932 年生。1955 年京都大学理学部物理学科卒業。京都大学基礎物理学研究所、名古屋大学工学部、九州大学工学部を経て、現在、九州工業大学情報工学部知能情報工学科教授。工学博士。知識情報処理、教育工学、自然言語処理およびその応用に関心を持っている。人工知能学会、電子情報通信学会、CAI 学会などの会員。

